Meet-in-the-Middle Attacks on Reduced-Round GOST

Gautham Sekar[*,1], Nicky Mouha[†,2,3], and Bart Preneel[‡,2,3]

[1]Indian Statistical Institute, Chennai Centre, SETS Campus, MGR Knowledge City, CIT Campus, Taramani, Chennai 600113, India.
[2]Department of Electrical Engineering ESAT/COSIC, KU Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium.
[3]iMinds, Belgium.

The block cipher GOST (GOST 28147-89) is a Russian standard for encryption and message authentication that is included in OpenSSL 1.0.0. In this paper, we present meet-in-the-middle attacks on several block ciphers, each consisting of 22 or fewer rounds of GOST. Our 22-round attack on rounds 10–31 requires only 5 known plaintexts and a computational effort equivalent to testing about 2^{223} keys for a success probability of $1 - 2^{-65}$. This attack is the best (going by the number of rounds) low data complexity key-recovery attack on GOST.

Keywords: Cryptanalysis, block cipher, meet-in-the-middle attack, Feistel network, GOST

1 Introduction

The GOST block cipher (GOST 28147-89) is a Russian standard for encryption and message authentication [1]. From hereon, we will refer to it as "GOST" for simplicity. It was designed in the erstwhile USSR, and declassified in 1989. This cipher is used in several applications, including OpenSSL 1.0.0, an open source toolkit for SSL/TLS [2].

[*]sgautham@isichennai.res.in
[†]Nicky.Mouha@esat.kuleuven.be
[‡]Bart.Preneel@esat.kuleuven.be

Attack	Ref.	# Rounds	Time	Data	Pr[Success]
MitM	**This paper**	**8**	$\mathbf{2^{127}}$	**3 KPs**	$\mathbf{1-2^{-65}}$
MitM	**This paper**	**9, 10**	$\mathbf{2^{159}}$	**3 KPs**	$\mathbf{1-2^{-33}}$
MitM	**This paper**	**11, 12**	$\mathbf{2^{191}}$	**4 KPs**	$\mathbf{1-2^{-65}}$
Differential	[5]	13	Not given	2^{51} CPs	Not given
MitM	**This paper**	**13, 14**	$\mathbf{2^{223}}$	**4 KPs**	$\mathbf{1-2^{-33}}$
MitM	**This paper**	**16**	$\mathbf{2^{223}}$	**5 KPs**	$\mathbf{1-2^{-65}}$
MitM	**This paper**	**22**	$\mathbf{2^{223}}$	**5 KPs**	$\mathbf{1-2^{-65}}$
Slide	[6]	24	2^{64}	$\approx 2^{64}$ KPs	Not given
Slide	[6]	30	$2^{253.7}$	$\approx 2^{64}$ KPs	Not given
Reflection	[7]	30	2^{224}	2^{32} KPs	Not given
Reflection-MitM	[4, 8]	32	2^{225}	2^{32} KPs	Not given

Table 1: Full-key recovery attacks on GOST; if explicitly stated in the original paper, success probabilities are given as well (KP: known plaintext, CP: chosen plaintext, MitM: meet-in-the-middle)

Both GOST and the US standard DES [3] are Feistel networks. GOST has 32 rounds, a block size of 64 bits and a key size of 256 bits. Following its release to the public, several cryptanalysis results were published. Full-key recovery attacks on GOST are listed in Table 1. In this table, we omitted related-key attacks. Recently, attacks on the full 32 rounds of GOST have appeared. In our table, we include the reflection meet-in-the-middle attack by Isobe et al. [4], Note that our attacks are the best low data complexity attacks on GOST.

The meet-in-the-middle attack. Let $\mathcal{M}$ and $\mathcal{K}$ denote the message space and the key space, respectively. Let A_K, $B_K : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{M}$ denote two block ciphers and let $Y_K = B_K \circ A_K$, where $\circ$ denotes function composition. In a meet-in-the-middle attack, the adversary deduces K from a known plaintext-ciphertext pair (p, c), where $c = Y_K(p)$, by solving $A_K(p) = B_K^{-1}(c)$.

In this paper, we use a variant of this technique to attack 16 rounds of GOST. In this approach, the place where the meet-in-the-middle occurs is at the subkeys instead of at the intermediate texts. This technique, which

we call *subkey meet-in-the-middle*, will be explained in Sect. 4. Our work is inspired by recent meet-in-the-middle attacks on XTEA [9].

Contribution of this paper. In this paper, we present meet-in-the-middle attacks on block ciphers consisting of up to 22 rounds of GOST. Our aim is to find out the maximum number of rounds that can be attacked given the following criteria:

1. The key is recovered with an information theoretically optimal probability of success indicated by the unicity distance [10].
2. The attack is in a non-related-key setting.
3. The attack works for the full key space (i.e., no classes of weak keys are used).
4. Very few known plaintext-ciphertext pairs (KPs) are required.

These criteria make the scenario very difficult from the point of view of the attacker. In Table 1, the 24-round and 30-round slide attacks require almost the entire codebook. For the 32-round differential attack, the entire codebook is required. The 30-round reflection attack and 32-round reflection meet-in-the-middle attack also require a large number of KPs when compared to our 22-round attack, whereas the time complexities are similar. Therefore, our 22-round attack may be regarded as the best attack (going by the number of rounds) to recover the key with a low data complexity. All the attacks in this paper have negligible memory requirements. This will be immediately apparent from the description of the attacks, because the intermediate calculations do not have to be retained in memory.

Biryukov and Wagner show in [11] that the reversal in the order in which the subkeys are used in the last 8 rounds, helps preclude slide attacks. We find that this reversal is responsible for many of the attacks (including the 22-round one) in this paper.

Organization. The paper is organized as follows. The specifications of GOST algorithm are given in Sect. 2. In Sect. 3, we describe our attacks

on block ciphers consisting of up to 14 rounds of GOST. Sections 4 and 5 describe our attacks on 16 and 22 GOST rounds, respectively. We suggest countermeasures and conclude the paper in Sect. 6.

2 Description of GOST

First, we introduce the following notation. Addition and subtraction modulo 2^{32} will be represented by $\boxplus$ and $\boxminus$ respectively. We will use $\oplus$ to denote exclusive-OR, $\lll$ for left rotation and $\|$ for concatenation.

The block cipher GOST has a block size of 64 bits and a key size of 256 bits. It is a 32-round Feistel network in which each round uses eight 4×4 S-boxes.

The 256-bit key K of GOST is divided into eight 32-bit subkeys $K_0, \ldots, K_7$. At every round, one of the 8 subkeys is selected according to a simple key schedule. The 32-bit subkey α_i used in round i, where $1 \leq i \leq 32$, is chosen from the set $\{K_0, \ldots, K_7\}$ according to the following rule:

$$\alpha_i \leftarrow \begin{cases} K_{i-1 \bmod 8} & \text{if } i \in \{1, \ldots, 24\} \ , \\ K_{32-i \bmod 8} & \text{if } i \in \{25, \ldots, 32\} \ . \end{cases} \tag{1}$$

In this paper, we will show that the reversal of the round-key order (in the last 8 rounds) is not a good design choice with respect to meet-in-the-middle attacks.

The 64-bit input to round i of GOST consists of two 32-bit parts L_{i-1} and R_{i-1}. For round 1, the plaintext p is used as input: $(L_0 \parallel R_0) \leftarrow p$. The input for round $i+1$ is computed iteratively from the input to round i as given by $L_i \leftarrow R_{i-1}$ and $R_i \leftarrow L_{i-1} \oplus (S(R_{i-1} \boxplus \alpha_i) \lll 11)$. We select α_i according to (1). The concatenated output from the 8 S-boxes of round i is denoted by $S(x)$, where x is split into 4-bit words.

The ciphertext c of GOST is produced by concatenating the two parts obtained after the 32nd round: $c \leftarrow R_{32} \parallel L_{32}$. A full description of the GOST block cipher is given in [1].

Note that [1] does not specify the S-boxes. Saarinen [12] has developed an attack with 2^{32} CPs to recover the S-boxes, assuming the attacker has

# Rounds	Rounds
8	18–25, 19–26, 20–27, 21-28, 22-29, 23-30, 24-31
9	18–26, 19–27, 20–28, 21-29, 22-30, 23-31
10	18–27, 19–28, 20–29, 21-30, 22-31
11	18–28, 19–29, 20–30, 21-31
12	18–29, 19–30, 20–31
13	18–30, 19–31
14	18–31

Table 2: All r-round reduced block ciphers ($8 \leq r \leq 14$) with unused subkeys

black box access to the encryption device, and can specify the key used to encrypt. As his attack works for any number of rounds, it can be used to turn each of the attacks in this paper into an attack with secret S-boxes. The attacker first performs Saarinen's chosen-key attack to recover the S-boxes. Following this, let us suppose that the attacker intercepts a communication involving an unknown/secret key. The attacker can now, with his knowledge of the S-boxes, use our meet-in-the-middle techniques to recover this key.

3 Unused Subkey Attacks on up to 14 Rounds of GOST

In this section, we show how to construct an attack on block ciphers consisting of r rounds of GOST, where $8 \leq r \leq 14$. In each of these block ciphers, at least one subkey is not used. Therefore, exhaustive search requires less than 2^{255} encryptions on average.

From (1), we obtain ciphers with unused subkey(s). Table 2 lists all these ciphers.

We now evaluate the data and time required for attacking the block ciphers listed in Table 2. Let us consider a block cipher in which s 32-bit subkeys, $1 \leq s \leq 4$, are not used.

Given one plaintext-ciphertext pair (p_0, c_0), with each key guess, the attacker checks whether

$$E_K^{(a \ldots a+r-1)}(p_0) = c_0 \ , \tag{2}$$

where $E_K^{(a...a+r-1)}$ denotes the r-round (rounds a to $a+r-1$) encryption using the k-bit key K, where $k = (256 - 32 \cdot s)$. One KP is not sufficient, because the key space ($2^{256-32 \cdot s}$ keys K) is larger than the ciphertext space (2^{64} ciphertext blocks). Therefore, the attacker requires more known plaintext-ciphertext pairs to determine the key K with sufficiently high probability. The number of KPs is denoted by n.

For every candidate k-bit key K, the attacker tests (2) using the first KP. If this equality is satisfied, the attacker uses a subsequent KP to check

$$E_K^{(a...a+r-1)}(p_j) = c_j \ , \tag{3}$$

where j is at most $n-1$. If one of the n equations (2), (3) is not satisfied, the candidate key K is incorrect and can be discarded.

Throughout this paper, we use the reasonable assumption that every block cipher under consideration has perfect confusion and diffusion properties as defined by Shannon [10]. If either the plaintext or the key, or both are changed, we assume that the corresponding ciphertext will be generated uniformly at random, independent from previously obtained ciphertexts.

With this assumption, each of the 64-bit conditions resulting from (2), (3) is satisfied with probability 2^{-64}. We now calculate the data and time complexities for our attacks. All time complexities are stated as the number of equivalent encryptions of the reduced-round block cipher.

The average success probability can be calculated as follows. The n 64-bit conditions are simultaneously satisfied with probability $2^{-n \cdot 64}$. The attacker can therefore eliminate a wrong key with probability $1 - 2^{-n \cdot 64}$. Assume that key m is the correct key, where $0 \leq m < 2^k$. This key will be found by our attack if all previous keys are eliminated. This happens with probability $(1 - 2^{-n \cdot 64})^m$. The correct key can be located anywhere among the list of 2^k candidate keys with equal probability. Therefore, the average success probability is

$$\begin{aligned} 2^{-k} \cdot \sum_{m=0}^{2^k-1} (1 - 2^{-n \cdot 64})^m &= 2^{n \cdot 64 - k} \cdot (1 - (1 - 2^{-n \cdot 64})^{2^k}) \\ &\approx 2^{n \cdot 64 - k} \cdot (1 - e^{-2^{k - n \cdot 64}}) \\ &\approx 1 - 2^{k - n \cdot 64 - 1} \ , \end{aligned} \tag{4}$$

s	n	k	**Average time complexity**	**Average success probability**
1	4	224	2^{223}	$1 - 2^{-33}$
2	4	192	2^{191}	$1 - 2^{-65}$
3	3	160	2^{159}	$1 - 2^{-33}$
4	3	128	2^{127}	$1 - 2^{-65}$

Table 3: Time complexities and success probabilities of attacks of Sect. 3 for several values of s and n

assuming $2^{k-n\cdot 64} \approx 0$. The approximations result from using the first and the second order Taylor approximations of e^x around 0. We now calculate the time complexity of the attack. For a candidate key K to be determined as wrong, the expected number of trials is $1 + 2^{-64} + \ldots + 2^{-(n-1)\cdot 64}$. The average (equivalent) number of encryptions of the algorithm is given by:

$$\begin{aligned} & 2^{-k} \cdot \sum_{m=0}^{2^k-1} \left(m \cdot (1 + 2^{-64} + \ldots + 2^{-(n-1)\cdot 64}) + n \right) \\ & = \frac{1}{2} \cdot \frac{1 - 2^{-n\cdot 64}}{1 - 2^{-64}} \cdot (2^k - 1) + n \ . \end{aligned} \tag{5}$$

Table 3 gives the average time complexities and the average success probabilities for various values of s ($= (256 - k)/32$) and n. The approximate number of plaintext-ciphertext pairs that are needed can also be calculated from Shannon's unicity distance [10] as $\lceil k/64 \rceil$.

4 Subkey Meet-in-the-Middle Attack on 16-Round GOST

In this section, we analyze the block cipher consisting of rounds 17–32 of GOST. We begin with the observation that K_7 is used consecutively in rounds 24 and 25.

Our attack assumes that the S-boxes are bijective. Note, however, that a similar attack works for non-bijective S-boxes, but then the computations of the time complexity and success probability become more involved.

Let $K = (K_0, K_1, K_2, K_3, K_4, K_5, K_6, X)$, where X is not relevant to the analysis because the attacker exhaustively searches over all subkeys except K_7. For every candidate key K, the attacker computes $E_K^{(17\ldots23)}(p_0)$, given a plaintext-ciphertext pair (p_0, c_0), and gets L_{23} and R_{23}. Similarly, the attacker computes $D_K^{(26\ldots32)}(c_0)$ and gets L_{25} and R_{25}. Using $\alpha_{24} = S^{-1}((L_{25} \oplus L_{23}) \ggg 11) \boxminus R_{23}$ and $\alpha_{25} = S^{-1}((R_{25} \oplus R_{23}) \ggg 11) \boxminus L_{25}$, the subkeys used in rounds 24 and 25 are obtained. If they are equal (for a wrong candidate key K, this happens with probability 2^{-32}),[1] the attacker sets $K_7 \leftarrow \alpha_{24} = \alpha_{25}$.

Then, using $n-1$ other plaintext-ciphertext pairs (p_j, c_j), $1 \leq j \leq n-1$, the attacker tests if $E_K^{(17\ldots32)}(p_j) = c_j$ with the value found for K_7. A wrong key will pass these tests with probability $2^{-32} \cdot \left(2^{-64}\right)^{n-1} = 2^{-32-(n-1)\cdot 64}$. Thus, with probability $1 - 2^{-32-(n-1)\cdot 64}$, a wrong key is eliminated. Using a similar reasoning as in Sect. 3, we obtain the average success probability:

$$
\begin{aligned}
& 2^{-224} \cdot \sum_{m=0}^{2^{224}-1} (1 - 2^{-32-(n-1)\cdot 64})^m \\
& = 2^{32+(n-1)\cdot 64-224} \cdot (1 - (1 - 2^{-32-(n-1)\cdot 64})^{2^{224}}) \\
& \approx 2^{32+(n-1)\cdot 64-224} \cdot (1 - e^{-2^{224-32-(n-1)\cdot 64}}) \\
& \approx 1 - 2^{224-32-(n-1)\cdot 64-1} \quad , \qquad (6)
\end{aligned}
$$

where the approximations hold when $n \geq 5$. We now calculate the time complexity of the attack. For a candidate key K to be determined as wrong, the expected number of trials is $1 + 2^{-32} + 2^{-32-64} + \ldots + 2^{-32-(n-2)\cdot 64}$. This is because for every candidate key K, the attacker always checks whether the subkeys used in rounds 24 and 25 agree. For 2^{-32} candidate keys, the attacker uses the second known plaintext, for 2^{-96} the attacker uses the third known plaintext, and so on. If the candidate key is correct, the attacker always performs n encryptions. As the correct key can be located anywhere in the list of 2^{224} candidates keys with equal probability, the

[1] If the texts obtained by encrypting p_0 and decrypting c_0, in the 13 outer rounds, are distributed uniformly at random, then so are the subkeys in rounds 24 and 25.

average number of 16-round computations is

$$2^{-224} \cdot \sum_{m=0}^{2^{224}-1} \left(m \cdot (1 + 2^{-32} + 2^{-32-64} + \ldots + 2^{-32-(n-2)\cdot 64}) + n \right)$$
$$= \frac{1}{2} \cdot (1 + 2^{-32} + 2^{-32-64} + \ldots + 2^{-32-(n-2)\cdot 64}) \cdot (2^{224} - 1) + n \ . \tag{7}$$

Substituting $n = 5$ in (6) and (7), the average success probability is $1 - 2^{-65}$ and the average number of 16-round computations is 2^{223}.

5 Attack on 22-Round GOST

From (1), we observe that the subkey K_0 is used only once in the block cipher consisting of rounds 10–31 of GOST. Therefore, here the attacker first checks for the equality of R_{16} and R'_{16}. These are obtained by respectively computing $E_K^{(10\ldots16)}(p_0)$ and $D_K^{(18\ldots31)}(c_0)$, where

$$K = (X, K_1, K_2, K_3, K_4, K_5, K_6, K_7) \ . \tag{8}$$

As subkey K_0 is not necessary to perform these partial encryptions and decryptions, X can be any 32-bit value.

If $R_{16} = R'_{16}$ (this happens with probability 2^{-32}), the corresponding value of K_0 ($= \alpha_{17}$) is obtained using:

$$\alpha_{17} = S^{-1}((R_{17} \oplus L_{16}) \ggg 11) \boxminus R_{16} \ . \tag{9}$$

The attacker then uses $n - 1$ KPs (p_j, c_j) to check $E_K^{(10\ldots31)}(p_j) = c_j$ with the value obtained for K_0. For every j, where j is at most $n - 1$, this equation is satisfied with probability 2^{-64}.

Using the same formulas as in Sect. 4, we find an average time complexity of 2^{223} for a success probability of $1 - 2^{-65}$. A similar attack can be mounted on other reduced-round block ciphers, each with less than 22 GOST rounds (e.g., rounds 11–31), where a particular subkey is used only once. Again, attacks similar to those in this section can be applied to the respective block ciphers even if the S-boxes are not bijective.

6 Conclusions and Open Problems

This paper presented several meet-in-the-middle attacks on GOST reduced to up to 22 rounds. To the best of our knowledge, the 22-round attack is the best attack (going by the number of rounds) to recover the key with very few known plaintexts.

Our attacks use different approaches – attacks on 14 or fewer rounds use a straightforward meet-in-the-middle approach and so does the 22-round attack; in the 16-round attacks, the meet-in-the-middle corresponds to inner round subkeys rather than middle text values. Our attacks work in a non-related-key setting.

The time complexity of both the 16-round and 22-round attacks is 2^{223}. It is required in these attacks that the S-boxes are bijective, but similar attacks can be constructed as well if this is not the case.

Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments and suggestions. This work has been funded in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II, in part by the Research Council KU Leuven: GOA TENSE, and in part by the Research Fund KU Leuven, OT/08/027.

References

[1] Pieprzyk, J. and Tombak, L. Soviet Encryption Algorithm, 1994. `http://freeworld.thc.org/root/phun/stego-challenge/gost-spec.pdf`.

[2] OpenSSL Software Foundation OpenSSL version 1.0.0, 2010. `http://www.openssl.org/`.

[3] National Institute of Standards and Technology FIPS PUB 46-3: Data Encryption Standard (DES), 1999. Supersedes FIPS 46-2.

[4] Isobe, T. A Single-Key Attack on the Full GOST Block Cipher. FSE, 2011, Springer Vol. 6733 of Lecture Notes in Computer Science, 290–305.

[5] Seki, H. and Kaneko, T. Differential Cryptanalysis of Reduced Rounds of GOST. Selected Areas in Cryptography, 2000, Springer Vol. 2012 of Lecture Notes in Computer Science, 315–323.

[6] Biham, E., Dunkelman, O., and Keller, N. Improved Slide Attacks. FSE, 2007, Springer Vol. 4593 of Lecture Notes in Computer Science, 153–166.

[7] Kara, O. Reflection Cryptanalysis of Some Ciphers. INDOCRYPT, 2008, Springer Vol. 5365 of Lecture Notes in Computer Science, 294–307.

[8] Dinur, I., Dunkelman, O., and Shamir, A. Improved Attacks on Full GOST. Cryptology ePrint Archive, Report 2011/558. `http://eprint.iacr.org/`.

[9] Sekar, G., Mouha, N., Velichkov, V., and Preneel, B. Meet-in-the-Middle Attacks on Reduced-Round XTEA. CT-RSA, 2011, Springer Vol. 6558 of Lecture Notes in Computer Science, 250–267.

[10] Shannon, C. E. Communication Theory of Secrecy Systems. Bell System Technical Journal, 28, 1949, 656–715.

[11] Biryukov, A. and Wagner, D. Advanced Slide Attacks. EUROCRYPT, 2000, Springer Vol. 1807 of Lecture Notes in Computer Science, 589–606.

[12] Saarinen, M.-J. A chosen key attack against the secret S-boxes of GOST, 1998. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.5532`.